

## Requisições de intervalo HTTP

Uma requisição de intervalo HTTP solicita que o servidor envie partes de um recurso de volta para um cliente. Requisições de intervalo são úteis para vários clientes, incluindo players de mídia que suportam acesso aleatório, ferramentas de dados que precisam apenas de parte de um arquivo grande, e gerenciadores de download que permitem aos usuários pausar e retomar um download.

### Verificando se um servidor suporta requisições parciais

Se uma resposta HTTP incluir o cabeçalho `Accept-Ranges` com qualquer valor diferente de `none`, o servidor suporta requisições de intervalo. Se as respostas omitirem o cabeçalho `Accept-Ranges`, isso indica que o servidor não suporta requisições parciais. Se requisições de intervalo não forem suportadas, aplicações podem se adaptar a essa condição; por exemplo, gerenciadores de download podem desabilitar botões de pausa que dependem de requisições de intervalo para retomar um download.

Para verificar se um servidor suporta requisições de intervalo, você pode emitir uma requisição `HEAD` para inspecionar os cabeçalhos sem solicitar o recurso completo. Se você usar `curl`, pode usar a opção `-I` para fazer uma requisição `HEAD`:

### BASH

```
bash
curl -I https://i.imgur.com/z4d4kwlk.jpg
```

Isso produzirá a seguinte requisição HTTP:

### HTTP

```
makefile
HEAD /z4d4kwlk.jpg HTTP/2
Host: i.imgur.com
User-Agent: curl/8.7.1
Accept: */*
```

Nesta resposta, `Accept-Ranges: bytes` indica que "bytes" pode ser usado como unidade para definir um intervalo (atualmente, nenhuma outra unidade é possível). O cabeçalho `Content-Length` também é útil, pois indica o tamanho total da imagem se você fosse fazer a mesma requisição usando o método `GET`.

### Solicitando um intervalo específico de um servidor

Se o servidor suporta requisições de intervalo, você pode especificar qual parte (ou partes) do documento deseja que o servidor retorne incluindo o cabeçalho `Range` em uma requisição HTTP.

### Intervalos de uma única parte

Podemos solicitar um único intervalo de um recurso usando `curl` para ilustrar. A opção `-H` adiciona uma linha de cabeçalho à requisição, que neste caso é o cabeçalho `Range` solicitando os primeiros 1024 bytes. A última opção é `--output -` que permite imprimir a saída binária no terminal:

### BASH

```
bash
curl https://i.imgur.com/z4d4kwlk.jpg -i -H "Range: bytes=0-1023" --output -
```

A requisição emitida se parece com isto:

### HTTP

```
makefile
GET /z4d4kwk.jpg HTTP/2
Host: i.imgur.com
User-Agent: curl/8.7.1
Accept: */*
Range: bytes=0-1023
```

O servidor responde com um status 206 Partial Content:

## HTTP

```
yaml
HTTP/2 200
content-type: image/jpeg
last-modified: Thu, 02 Feb 2017 11:15:53 GMT
...
accept-ranges: bytes
content-length: 146515
```

O cabeçalho Content-Length indica o tamanho do intervalo solicitado, não o tamanho total da imagem. O cabeçalho de resposta Content-Range indica onde esta mensagem parcial pertence dentro do recurso completo.

## Intervalos de múltiplas partes

O cabeçalho Range também permite obter múltiplos intervalos de uma só vez em um documento multipart. Os intervalos são separados por uma vírgula.

O servidor responde com o status 206 Partial Content como mostrado abaixo. A resposta contém um cabeçalho Content-Type, indicando que um multipart/byterange segue. A string de delimitação (neste caso 3d6b6a416f9b5) separa as partes do corpo, cada uma com seus próprios campos Content-Type e Content-Range:

## BASH

```
bash
curl http://www.example.com -i -H "Range: bytes=0-50, 100-150"
```

## HTTP

```
php-template
HTTP/1.1 206 Partial Content
Content-Type: multipart/byteranges; boundary=3d6b6a416f9b5
Content-Length: 282

--3d6b6a416f9b5
Content-Type: text/html
Content-Range: bytes 0-50/1270

<!doctype html>
<html lang="en-US">

--3d6b6a416f9b5
Content-Type: text/html
Content-Range: bytes 100-150/1270

<meta http-equiv="Content-type" content="text/html; c

--3d6b6a416f9b5--
```

## Requisições de intervalo condicionais

Ao retomar a solicitação de mais partes de um recurso, é necessário garantir que o recurso armazenado não foi modificado desde que o último fragmento foi recebido.

O cabeçalho HTTP If-Range torna uma requisição de intervalo condicional: se a condição for satisfeita, a requisição de intervalo será emitida e o servidor enviará de volta uma resposta 206 Partial Content com o corpo apropriado. Se a condição não for satisfeita, o recurso completo é enviado de volta, com um status 200 OK. Este cabeçalho pode ser usado com um validador Last-Modified ou com um ETag, mas não com ambos.

## HTTP

```
yaml
If-Range: Wed, 21 Oct 2015 07:28:00 GMT
```

## Respostas de requisição parcial

Existem três status relevantes ao trabalhar com requisições de intervalo:

- Uma requisição de intervalo bem-sucedida resulta em um status 206 Partial Content do servidor.
- Uma requisição de intervalo fora dos limites resultará em um status 416 Requested Range Not Satisfiable, significando que nenhum dos valores de intervalo se sobrepõe à extensão do recurso. Por exemplo, o first-byte-pos de cada intervalo pode ser maior que o comprimento do recurso.
- Se requisições de intervalo não forem suportadas, um status 200 OK é retornado e todo o corpo da resposta é transmitido.

## Comparação com Transfer-Encoding *chunked*

O cabeçalho Transfer-Encoding permite a codificação em partes (chunked), que é útil quando grandes quantidades de dados são enviadas ao cliente e o tamanho total da resposta não é conhecido até que a requisição tenha sido completamente processada. O servidor envia dados ao cliente imediatamente, sem armazenar em buffer a resposta ou determinar o comprimento exato, o que leva a uma latência melhorada. Requisições de intervalo e codificação chunked são compatíveis e podem ser usadas juntas ou separadamente.

## Veja também

- Códigos de status relacionados: 200, 206, 416.
- Cabeçalhos relacionados: Accept-Ranges, Range, Content-Range, If-Range, Transfer-Encoding.